

REMARKS

Claims 1-17 are pending. Claims 1-4, 6, 7, 10, 12-15 and 17 have been amended.

Claims 1-17 stand rejected under 35 U.S.C. § 112, ¶ 2 for various reasons. Claims 1 and 12-17 further stand rejected under 35 U.S.C. § 101. Claims 1-17 also stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,721,727.

Reconsideration is respectfully requested.

The Section 112 Rejections

Although Applicants disagree that the claims as originally presented are not clear, Applicants nevertheless have amended claims 1-4, 6, 7, 10, 12-15 and 17 in the hope of moving prosecution forward. Applicants respectfully submit that claims 1-17, as amended, fully comply with the requirements of Section 112. Reconsideration of the Section 112 rejections is respectfully requested.

The Section 101 Rejections

Applicants have amended claims 1 and 12. Applicants submit that these amendments overcome the bases for the Section 101 rejections. Reconsideration is respectfully requested.

The Section 102(e) Rejection

Claims 1-17 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,721,727 ("Chau"). Applicants respectfully submit that the present claims recite features that are neither taught nor suggested by Chau.

The present invention is directed to a method that enables XML data to be stored in a field of a user defined type, while other fields of the user defined type can store other types of data. For example, as illustrated in Figure 5, a user defined type called "Employee" can be created having fields called fName, lName, Salary, Age and Resume. An instance of the Employee user defined type can thus be used to store an employee's name, salary, age and resume. The fName, lName, Salary and Age fields are defined in this example as having the traditional string, floating point (*i.e.*, "double") and integer data types, respectively, whereas the Resume field is defined to hold XML data.

The ability to store XML data in a field of a user defined type is achieved, at least in part, by first defining a new XML data type as a class in managed code. In the present embodiment, this new class is called “SqlXml”. *Specification*, ¶¶ 46-50. Then, a user defined type can be created in which at least one of the fields of the user defined type is defined as having the new XML data type, *i.e.*, the field is defined as an instance of the SqlXml class. Referring again to the example illustrated in Figure 5, the “Employee” user defined type has a field called “Resume” that is defined as having the new “SqlXml” type. That field of the user defined type can thus be used to store an XML document, *e.g.*, an employee’s resume. Thus, the present invention enables XML data to be stored in a field of an instance of a user defined type, while other fields of the user-defined type store other types of data.

These features are recited in each of the independent claims. For example, claim1 (as amended) recites in part:

A method for use in a database system in which a user defined type is defined by a class in managed code and comprises one or more fields, each field having a respective data type, the method comprising:

defining another class in managed code that represents an XML data type;

defining at least one of the fields within the managed code class that defines the user defined type, as an instance of the managed code class that represents the XML data type

....

Independent claims 7 and 12 recite similar features. With the present invention, a user can then create an instance of the user defined type and store that instance in a database. Chau does not teach or suggest defining a class in managed code that represents an XML data type and then defining at least one field of a user defined type as an instance of the class that represents the XML data type.

On the contrary, as the title of Chau indicates, Chau merely describes a technique for storing XML documents in a column of a relational database table. Applicants acknowledged such techniques in the “Background” section of the instant application:

Existing database management systems provide support for storing XML data in a relational database store. For example,

Microsoft's SQL SERVER provides support for XML data type *columns*, variables and parameters. *You can create a table with one or more XML columns, store XML values in the XML columns, type an XML column using an XML schema namespace, index the XML column, and query into the XML values.*

Specification, ¶ 21 (emphasis added).

As Applicants further pointed out, however:

while it has been possible in the past to store XML data in a relational data base in these instances, it would be desirable to be able to embed XML data in a field of a user defined type that is created in managed code. The present invention provides this ability.

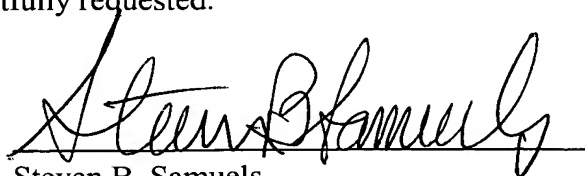
Thus, Chau merely describes a technique for storing XML documents in a column of relational database table. Chau does not teach or suggest any method for storing XML data in a field of a user defined type, as does the method claimed by Applicants.

Because Chau does not teach or suggest the claimed invention, Applicants respectfully submit that independent claims 1, 7 and 12, as amended, patentably define over Chau and any other art of record, alone or in combination. Inasmuch as the other claims all depend either directly or indirectly from one of these independent claims, Applicants submit that they too patentably define over the art of record. Reconsideration of the Section 102(e) rejection of claims 1-17 is therefore respectfully requested.

CONCLUSION

Applicants respectfully submit that the instant application is now in condition for allowance. An early Notice of Allowance is respectfully requested.

Date: June 16, 2006



Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439